[0226] A constructor accessor of the present invention is an operation accessor and a feature accessor on a model accessor. Constructor accessors provide a mechanism for creating new instances of the model accessed by the model accessor.

[0227] A constructor accessor has a one-to-one association relationship with a constructor descriptor that is the constructor descriptor for which the constructor accessor provides access to an implementation. A constructor accessor has a one-to-one association relationship with a constructor implementation that is the constructor implementation that this accessor uses to perform the actual construction. The parameter accessors represent the parameters in the constructor. Constructor accessors inherit all the attributes from feature accessor and operation accessor, but do not add any additional attributes.

[0228] A constructor accessor includes the following operation: newInstance(ParameterList) that is the method call to perform a construction with the given values as parameters. The parameters in the parameter list must match the type and be accepted by the parameter accessors in order for construction to be successful.

[0229] A constructor accessor adds no additional events.

[0230] A destructor accessor of the present invention is an operation accessor and a feature accessor of a model accessor. Destructor accessors provide a mechanism to destroy an instance of the model being accessed by the model accessor. Depending on the persistence rules associated with a meta-accessor implementation, this may permanently destroy the instance from the persistence store, or it may simply remove the representation of that persisted object from memory. The accessor implementer will document the level of destruction as it applies to persisted data.

[0231] A destructor accessor has a one-to-one association relationship with a destructor descriptor that is the destructor descriptor for which this accessor provides access to an implementation. A destructor accessor has a one-to-one association relationship with a destructor implementation that is the destructor implementation that the destructor accessor uses to perform the actual destruction. A destructor accessor has a zero-to-many association relationship with precondition constraints that are each an implementation of a constraint to restrict some environmental condition or state of the model implementation before the execution of the operation. These types of constraints might check for the installation of a security manager, the existence of a database connection, or some other environmental condition. These constraints might also check to see if a model implementation has entered into the correct state to allow the execution of this operation. Constraints related to the parameter values are held by the parameter descriptors.

[0232] A destructor accessor includes the following operation: destroy( ) that is the method call to perform a destruction.

[0233] A destructor accessor adds no additional events.

[0234] A model accessor of the present invention is an accessor that defines the responsibilities for accessing an implementation of a model corresponding to a specific metamodel. For each feature in the metamodel, there must be a feature accessor in the model accessor.

[0235] A model accessor has a one-to-one association relationship with a metamodel that the metamodel for which the model accessor provides access to an implementation. There exists a one-to-one relationship for each descriptor in the metamodel to each child accessor in this model accessor. A model accessor has a one-to-one association relationship with a parent model accessor that the parent model accessor is configured to give access to all the features of the parent model. This model accessor delegates to its parent model accessor wherever the model it describes would use features inherited from it parent. A model accessor has a one-to-one association relationship with a version that provides details about the number of modifications that have been made to the model implementation. A model accessor has a one-to-many aggregation relationship with constructor accessors that provide a mechanism for creating new instances of the model described by the model descriptor. There may exist several different constructors, each of which uses a different number and type of arguments. A model accessor has a one-to-one aggregation relationship with a destructor accessor that provides a mechanism to destroy an instance of the model being described. Depending on the persistence rules associated with a meta-accessor implementation, this may permanently destroy the instance from the persistence store, or it may simply remove the representation of that persisted object from memory. The accessor implementer will document the level of destruction related to persisted data. A model accessor has a zero-to-many association aggregation with static attribute accessors that are attribute accessors that do not require an instance of the model in order to be accessed. Static attribute accessors allow the value to be set and retrieved where the attribute is shared among all instances of the model. A model accessor has a zero-to-many aggregation relationship with instance attribute accessors that are attribute accessors that require an instance of the model in order to be accessed. Instance attribute accessors allow the value to be set and retrieved on one specific instance. A model accessor has a zero-to-many aggregation relationship with static operation accessors that do not require an instance of the model in order to be executed. Static operation accessors provide a mechanism to execute the operation. A model accessor has a zero-to-many association relationship with instance operation accessors that are operation accessors that require an instance of the model in order to be executed. Instance operation accessors provide a mechanism to execute the operation. A model accessor has a zero-to-many association relationship with signal accessors that provide the mechanism to register an appropriate instance's interest in receiving notification when the model generates an event. Signal accessors the mechanism to remove interest in receiving notification for an event. The instance registering interest in receiving event notification must implement the appropriate interface to match the listener type described in the signal descriptor.

[0236] A model accessor includes the following operation: newInstance(ParameterList) that is used to construct a new instance of the model. This operation delegates to the appropriate constructor accessor to return the appropriate instance.

[0237] To register interest in events fired by an instance of a model implementation, interested listeners should register using the signal accessors. To register interest in signals fired by the model accessor, a listener should use registerImplementationListener(listener) inherited from the accessor base